Subject:
Date

تکلیف چهارم طراحی الگوریتم — محمد شاملو — ۸۱۰۱۰۰۰۸۴ — استاد اسدپور

۱) * ابتدا با یک دور پیمایش و برای پیش پردازش از $O(n)$ ، max مقادیر آرایه را مشخص می کنیم.
سپس آرایه ای به طول این max می سازیم (A)
سپس با یک پردازش دیگر روی همه ی اعضای آرایه با رسیدن به هر عدد در آرایه مثل a یکی به مقدار $A[a]$ اضافه می کنیم.
مشخصا در این راه حل آرایه ی اصلی به طول n، دو بار پیمایش شده است پس دو بار از $O(n)$ وقت انجام شده که معادل آن است که الگوریتم از $O(n)$ باشد!
مشکل این راه حل نیاز به حافظه بسیار زیاد است!

* ابتدا کل آرایه را از $O(n\log n)$ order سورت می کنیم در این حالت با یک بار حرکت روی اعضای آرایه و مقایسه ی دو عنصر متوالی می توانیم مسئله را حل کنیم.
به این ترتیب که در صورتی که عدد بعدی با عدد فعلی مساوی شود متغیر count را یکی زیاد کنیم و در صورتی که متفاوت بود یک متغیر جدید count برای آن عدد جدید تعریف کرده و تعداد عنصر جدید را در آن ذخیره می کنیم.
این روش از $O(n\log n)$ و استفاده از حافظه بسیار کمتری است ✓

* همچنین مشخصا با روش bruit force و دو حلقه ی تو در تو که یکی یک عنصر را انتخاب و دیگری تعداد تکرار آن را در آرایه می شمارد می توان از $O(n^2)$ مسئله را حل کرد!

PAPCO

2) ابتدا به‌عنوان پیش‌پردازش آرایه را با استفاده از الگوریتم‌های سریع مرتب‌سازی، مرتب می‌کنیم (O(nlogn))

سپس هر عنصر را با عنصر بعدی خودش مقایسه می‌کنیم و درصورتی که تفاضل آنها یک باشد آن را حذف می‌کنیم و آن جفت را حذف می‌کنیم!

پس اردر نهایی ما از O(nlogn) خواهد بود

همچنین مشخص است اگر برای هر عنصری که در آرایه بدانیم که حداقل دو عنصر با تفاضل کوچکتر مساوی یک وجود داشته باشد می‌توان به انتها رسید اما در غیر این صورت ممکن نیست!

A = [۱, ۱, ۲, ~~۳~~ ۳, ۴, ۴, ۵, ۶] → [۶] ✓

B = [۱, ۱, ۲, ۴, ۴, ۵, ۶, ۶] → [۴, ۶] ×

Subject:
Date

(3

$$
\begin{aligned}
7x + 5y - 3z &= 16 \\
3x - 5y + 2z &= -8 \\
5x + 3y - 7z &= 0
\end{aligned}
\longrightarrow
\begin{array}{c} R_1 \\ R_2 \\ R_3 \end{array}
\begin{bmatrix} 7 & 5 & -3 \\ 3 & -5 & +2 \\ 5 & 3 & -7 \end{bmatrix} , \begin{bmatrix} 16 \\ -8 \\ 0 \end{bmatrix}
$$

در این مرحله باید سه عنصری که دور آن‌ها دایره رسم کرده‌ام را صفر کنیم. تا به ماتریس بالا مثلثی برسیم!

$$
\left[\begin{array}{ccc|c} 7 & 5 & -3 & 16 \\ 3 & -5 & 2 & -8 \\ 5 & 3 & -7 & 0 \end{array}\right]
\xrightarrow{7R_3 - 5R_1}
\left[\begin{array}{ccc|c} 7 & 5 & -3 & 16 \\ 3 & -5 & 2 & -8 \\ 0 & -4 & -34 & -80 \end{array}\right]
\xrightarrow{7R_2 - 3R_1}
$$

$$
\left[\begin{array}{ccc|c} 7 & 5 & -3 & 16 \\ 0 & -50 & 23 & -104 \\ 0 & -4 & -34 & -80 \end{array}\right]
\xrightarrow{25R_3 - 2R_2}
\left[\begin{array}{ccc|c} 7 & 5 & -3 & 16 \\ 0 & -50 & 23 & -104 \\ 0 & 0 & -896 & -1792 \end{array}\right]
$$

$$\Rightarrow \boxed{z = 2} \longrightarrow -50y + 46 = -104 \Rightarrow \boxed{y = 3}$$

$$\Rightarrow 7x + 15 - 6 = 16 \Rightarrow \boxed{x = 1}$$

4) از نقطه‌ی x یک نیم خط با مبدأ x رسم می‌کنیم و تعداد نقاط برخورد نیم خط رسم شده با شکل اولیه را می‌شماریم در صورتی که تعداد برخوردها فرد باشد نقطه x داخل شکل بوده و در غیر این صورت x خارج شکل است! علت این مسئله آن است که در صورتی که x خارج از شکل باشد با ورود هر نیم خط به شکل حتماً باید از شکل خارج هم بشد چون شکل نامتناهی نیست! پس اگر x خارج از شکل باشد به ازای هر برخورد ورود به شکل یک برخورد خروج از شکل نیز داریم که باعث می‌شود تعداد برخوردها زوج تا باشد.

5) قصد داریم $P(x_0)$ را تعیین کنیم.

a_k	a_{2n+1}	a_{2n-1}	a_{2n-3}	---	a_3	a_1
$b_{k+1} \times (x_0)$	0	$a_{2n+1} \times (x_0)$				
b_k	a_{2n+1}	$a_{2n+1}x_0 + a_{2n-1}$				b_0

جدول هورنر

به ازای هر ضریب از $P(x)$ یک عملیات جمع و یک عملیات ضرب باید انجام دهیم پس با یک حرکت روی تمامی ضریب‌ها و جمع و ضرب با عنصر قبلی و ... مطابق شکل می‌توان $P(x_0)$ را محاسبه کرد! که این کار نیاز به یکبار دیدن هر ضریب است یعنی از $O(n)$ قابل حل است!

6) از عنصر اول آرایه شروع می‌کنیم (عنصر ۱) ریشه Heap است! سپس عنصر را با عنصر ۲ام و
۳ام که فرزندانش هستند مقایسه می‌کنیم اگر عنصر اول بزرگتر بود یعنی یک max-heap داریم حال
بقیه عناصر را بررسی می‌کنیم اگر اندیس هر عنصر را i لحاظ کنیم تا عنصر وسط آرایه پیمایش روی
می‌کنیم و اگر همه‌ی این عناصر از عنصر 2i و 2i+1 ام بزرگتر بودند یعنی max-heap داریم
همچنین اگر عنصر اول از عنصر دوم و سوم کمتر بود یعنی باید min-heap بودن بررسی شود که با یکبار
پیمایش آرایه و بررسی هر عنصر با اندیس i با عنصر 2i و 2i+1 به عنوان فرزندانش
اگر آنها از عنصر i بزرگتر بودند و عنصر i کمتر بود و این برای همه‌ی عناصر ما قبل وسط
آرایه صادق بود ما یک min-heap داریم در غیر این صورت آرایه نمایش دهنده هیچ heap نیست

پیراهن → X
دامن → Y
شلوار → Z ⟹ سود $= 6X + 4Y + 8Z$

محدودیت‌ها:

طراحی: $\frac{X}{100} \times 3 + \frac{Y}{100} \times 2 + \frac{Z}{100} \times 2 \leq 12$

برش: $\frac{X}{100} \times 1 + \frac{Y}{100} \times 2 + \frac{Z}{100} \times 3 \leq 14$

دوخت: $\frac{X}{100} \times 2 + \frac{Y}{100} \times 4 + \frac{Z}{100} \times 4 \leq 16$

با حفظ شرایط فوق باید سود را که مطابق فرمول محاسبه می‌شود max کنیم.

8) پایه استقراء: $K=1$ ، $A^1_{ij}=1$ ← (درصورتی که v_i و v_j همسایه باشند مسیر داریم) if $(v_i, v_j) \in E$

$A^1_{ij}=0$ ← (درصورتی که v_i و v_j همسایه نباشند مسیر به طول ۱ قطعاً نداریم) if $(v_i, v_j) \notin E$

فرض استقراء: $K-1$: A^{K-1}_{ij} تعداد مسیرهای به طول $K-1$ بین v_i و v_j را مشخص می‌کند.

حکم استقراء: باید ثابت کنیم A^K_{ij} تعداد مسیرهای بین v_i و v_j در طول K را نشان می‌دهد

$$A^K_{ij} = (A^{K-1}_{ij} \cdot A_{ij}) = \sum_{m=1}^{n} A^{K-1}_{im} \cdot A_{mj}$$

یعنی تعداد مسیرهای به طول $K-1$ بین v_i و v_m به اضافه مسیر به طول ۱ از v_m به v_j با ضرب کردن این نقطه واسط در بین مسیر نهایتاً $(K-1+1)$ تعداد مسیرهای بین v_i و v_j خواهد بود

طبق فرض استقراء A^{K-1}_{im} می‌دانیم مسیرهای به طول $K-1$ بین m و i را به ما می‌دهد
چون تمامی نقاط به عنوان m فرض می‌شود و مسیر به طول ۱ از m به j نیز شمرده‌ش می‌شود
برای یک m وجود داشته باشد و هر دو صفر نباشد تعداد مسیر از i به j پیدا می‌شود که معادل A^K است
حکم استقراء ثابت شد پس حکم مسئله ثابت می‌شود